UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/900,123	07/05/2001	Scott Wiltamuth	MSFT-0573/160076.1	5765

41505   7590   04/28/2009
WOODCOCK WASHBURN LLP (MICROSOFT CORPORATION)
CIRA CENTRE, 12TH FLOOR
2929 ARCH STREET
PHILADELPHIA, PA 19104-2891

EXAMINER
VO, TED T

ART UNIT	PAPER NUMBER
2191	

MAIL DATE	DELIVERY MODE
04/28/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

	Application No.	Applicant(s)
Office Action Summary	09/900,123	WILTAMUTH ET AL.
	Examiner	Art Unit	
	TED T. VO	2191	

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) ☒ Responsive to communication(s) filed on <u>13 February 2009</u>.

2a) ☐ This action is **FINAL**.   2b) ☒ This action is non-final.

3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) ☒ Claim(s) <u>1-3,6-15,17-19,23-26,28-37,39-41,58-63 and 66-75</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) <u>1-3,6-15,17-19,23-26,28-37,39-41,58-63 and 66-75</u> is/are rejected.

7) ☐ Claim(s) _____ is/are objected to.

8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) ☐ The specification is objected to by the Examiner.

10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a) ☐ All   b) ☐ Some * c) ☐ None of:

      1. ☐ Certified copies of the priority documents have been received.

      2. ☐ Certified copies of the priority documents have been received in Application No. _____.

      3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
    Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
    Paper No(s)/Mail. Date. _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

DETAILED ACTION

1.   This action is in response to the communication filed on 02/13/2009.

       Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, 66-75 are pending in the

application.

Claim Rejections - 35 USC § 112

2.   The following is a quotation of the second paragraph of 35 U.S.C. 112:

       The specification shall conclude with one or more claims particularly pointing out and
       distinctly claiming the subject matter which the applicant regards as his invention.

3.   Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, 66-75 are rejected under 35

U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and

distinctly claim the subject matter which applicant regards as the invention.

As per claims 1-3, 6-15, 17-19: Claims appear being a method for use. The scope of these claims

is unclear. The claims mix the method for use with a description of a high level program syntax

('an explicit interface member'). The method for use appear being done by a user for how to

configure a computer by utilizing mechanism to implement a class and then to store the class in

the class into a readable mechanism. The method is mixed with a programming syntax that is

unrelated to the use of the computer, or the method for use. Therefore, it is unclear whether the

language of claims 1-3, 6-15, 17-19 is directed to programming language syntax or using a

computer system. A court decision has determined that a claim directed to a system and a

method for using that system is indefinite. See IPXL Holdings, 430 F.3d at 1384. See also $3

Inc., 259 F.3d at 1372 ("When the claims become so ambiguous that one of ordinary skill in the

art cannot determine their scope absent speculation, such claims are invalid for indefiniteness.")

(citing In re Steele, 305 F.2d at 862-63).

Claim 11 further recites Java which is trade name without using the trademark label.

Furthermore, it is unclear the limitation of the claim is true or not; because many programming

languages appear not being object-oriented programming language.

The interpretation for the claims is configuring a computer by utilizing a mechanism to

store a class in the computer readable medium.

As per claims 23-26, 28-37, 39-41: The claims recite "stored thereon a plurality of computer-

executable modules written in an object-oriented programming language". This limitation is

ambiguous. In light of the specification, the object-oriented programming language appears C#;

and therefore is no mechanism that shows this object-oriented programming language can be

executable at a high level module. The evidence is shown as being admitted in the recitation in

the claim 33. "*A computer readable storage medium according to claim 23, wherein the object-*

oriented programming language is one of C#, Fortran, Pascal, Visual Basic, C, C++ and Java".

The claim has the Java trade name, it should be labeled with the trademark, and it should be also

noted that there is no processor executing a high level language program. Such as a Fortran

program cannot be executed by a processor. Interpret the modules are high level language

programs.

<u>As per claims 58-63, 66-75</u>: Claims recite method for generating an object by using a compiler, in other words, to instantiate an object from a class by using a compiler. However, this method is unclear, whether it is for "generating an object" by the preamble or describing an explicit interface member. It should be noted that to include an explicit interface member does not make the method of generating an object different from the generation of an object without the explicit interface member. The claims look like describing a high level language syntax than generation of an object. The evidences are seen in the recitations of claim 58-60, 62-63, 66-75. For example: Claim 58 recites "wherein it is impossible…", it is not understandable the subject mater of the claim; claim 59 recites, "is permitted..", it cannot be functionality for thing that is permitted; Claim 60 recites, "includes.."; it is functionless to the method and is unclear since the inclusion is functionless to the method; ….; claim 74 recites "locating implemented interface", it does nothing to the method of the claim 61, and so on.

Claim Rejections - 35 USC § 101

4.   35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

5.   The claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41 are rejected under 35 U.S.C 101

because the claimed invention is directed to non-statutory subject matter.

<u>Claims 1-3, 6-15, 17-19</u>: The claims are merely the syntax description of a class in a high level

programming. The claims force themselves as storing the class to a computer readable medium.

This tends to show the claim as being a common method for using a computer to store a thing

into a computer readable medium without show any any practical application or a transformation

of a subject mater. It is known that with an operable computer; it is so easily for a user to store a

thing. There is not practical application in the claims. Thus, the claims fail to meet statutory

claims.

<u>Claims 23-26, 28-37, 39-41</u>: The claims recite a computer readable storage medium. Under a

statutory medium, it requires the medium being the physical medium and tangible in a computer,

and storing executable instruction so that when executable by a processor to do a practical thing.

The storage medium of the claim stores only modules written in high level programming. The

word "executable" clearly renders the claims indefinite. It is unclear whether a C++, pr Java

module is executable or not, but a C module, or a Fortran module cannot be executable because

the are at high level source code that cannot recognized by any computer processor.

Claim Rejections - 35 USC § 103

6.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> A person shall be entitled to a patent unless –
>
> (a) A patent may not be obtained though the invention is not identically disclosed or
> described as set forth in section 102 of this title, if the differences between the subject
> matter sought to be patented and the prior art are such that the subject matter as a whole
> would have been obvious at the time the invention was made to a person having ordinary
> skill in the art to which said subject matter pertains. Patentability shall not be negatived
> by the manner in which the invention was made.

7.   Claims 1-3, 6-15, 17-19, 23-26, 28-37, 39-41, 58-63, 66-75 are rejected under 35

U.S.C. 103(a) as being unpatentable over Kramer et al., "Configuring Object-based Distributed

Programs in REX" , Software Engineering Journal, 3-1992, pages 139-149.

<u>As per claim 1</u>: Kramer teaches

A method for <u>configuring</u> a computer <u>to generate computer executable instructions</u>, using

object-based computer code of an object-oriented programming language, the method

comprising:

utilizing an explicit interface member mechanism that enables a class to implement an explicit

interface member by explicitly specifying the relationship between the class and the explicit

interface member, wherein the explicit interface member mechanism enables an implemented

explicit interface member to be excluded from a public interface of said class; and

storing said class in a form that includes said implemented explicit interface member in a

computer readable storage medium.

- The claim, as a whole, is for <u>utilizing an explicit interface member mechanism</u> that <u>enables a</u>

<u>class</u> to implement an explicit interface member, and storing the class into a computer readable

storage medium. Thus, it is very common; particularly, in the area of the storage device, where a

storage device is for storing data such as software.

 The reference teaches using a mechanism (See p. 140, title) that is well defined with interfaces

(*explicit interface member mechanism*) for configuring a computer to generate programming

class (e.g. Fig. 4, p. 142). With this configuration language, in Fig. 4, it enables an object with

explicit interfaces (*an explicit interface member*). Using the configuration language, the object

with explicit interfaces will not require declaring "public" to the interfaces.

The reference does not address storing the class or the object with explicit interfaces as

emphasized in the claim with the feature "***storing said class in a form that includes said***

implemented explicit interface member in a computer readable storage medium".

However, this is the work done in for computer and in computer.

Therefore, it is obviously to ordinary in the art at the time to implement the text of this class into

a file for being portable, where according to MPEP, make portable cannot be patentable over a

reference.

<u>As per claim 2</u>: Kramer further teaches

A method according to claim 1, wherein said specifying of the relationship includes specifying

a qualified name of the class. See p. 139, parent child class, p. 140, object naming, etc.

As per claim 3: Kramer further teaches ***A method according to claim 2, wherein said specifying of the qualified name includes specifying an interface name and said at least one interface member name.*** See p. 139, parent child class, and further see Figure 1, or 4.

As per claim 6: Kramer further teaches *A method according to claim l, wherein the explicit interface member mechanism **enables** the class to implement an internal interface not accessible to a consumer of said class*. The bold phase is relative term that has the claim remain reciting "*explicit interface member mechanism*". See sec. 2.2, start at p. 141, and refer to the type of ODD specified in the reference title.

As per claim 7: Kramer further teaches *A method according to claim l, wherein said explicit interface member mechanism **enables disambiguation** of a plurality of interface members having the same signature.* The bold phase is relative term that has the claim remain reciting "*explicit interface member mechanism*". Inherent in the term "explicit interface" as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 8: Kramer further teaches *A method according to claim 1, wherein said explicit member mechanism **enables disambiguation** of a plurality of interface members having the same signature and return type.* The bold phase is relative term that has the claim remain reciting "*explicit interface member mechanism*". Inherent in the term "explicit interface" as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 9: Kramer further teaches *A method according to claim 1, wherein in addition to allowing the implementation of public interface members, said explicit interface member mechanism **enables the implementation** of private interface members.* The bold phase is relative

term that has the claim remain reciting "*explicit interface member mechanism*". Inherent in the

term "explicit interface" as shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 10: Kramer further teaches *A method according to claim 1, wherein said explicit*

*interface member mechanism **enables** the implementation of a plurality of non-conflicting*

specific versions of a generic interface. The bold phase is relative term that has the claim remain

reciting "*explicit interface member mechanism*". Inherent in the term "explicit interface" as

shown in Fig. 4, and see sec. 2.2, 2.3.

As per claim 11: Kramer further teaches *A method according to claim 1, wherein the computer*

code is programmed according to an object-oriented programming language, and said object-

*oriented programming language is **one of C#, Fortran, Pascal, Visual Basic, C, C++ and Java.***

The bold phase is only a programming type language, which has the claim remain reciting

"*object-oriented programming language*". Inherent in the term "class" in the reference, or see

the title.

As per claim 12: Kramer further teaches *A method according to claim 1, wherein an*

implementation of an explicit interface member is a method, property, event, or indexer

declaration that references a fully qualified interface member name. See sec. 2.2, 2.3.

As per claim 13 Kramer further teaches

A method according to claim 1, wherein the class names an interface in a base class list of the

class that contains a member whose fully qualified name, type, and parameter types exactly

match those of the implementation of the explicit interface member. See sec. 2.2, 2.3, and refer

to the used tern "object-based" as seen in the title.

As per claim 14: Kramer further teaches *A method according to claim 1, wherein said explicit*

interface member mechanism includes an interface mapping mechanism that locates

implementations of interface members in the class. See Fig. 5, or refer to object binding.

As per claim 15: Kramer further teaches *A method according to claim 14, wherein said*

interface mapping mechanism locates an implementation for each member of each interface

specified in a base class list of the class. See Fig. 5, or refer to object binding.

As per claim 17: Kramer further teaches *A method according to claim 1, wherein it is not*

possible to override an explicit interface member implementation, but where an explicit

interface member implementation calls another virtual method, derived classes are capable of

overriding the implementation. The limitation is indefinite. See sec. 2.2.

As per claim 18: Kramer further teaches *A method according to claim 1, wherein the class*

inherits an interface implementation ***is permitted*** *to re-implement the interface by including the*

interface in the base class list of the software component. The bold phase is a relative term,

which has the claim remain reciting the inheritance of object-oriented. See sec. 2.4.

As per claim 19: Kramer further teaches *A method according to claim 1, wherein said explicit*

interface member mechanism ***prevents conflict*** *among specific implementations of a generic*

interface. The bold phase is a relative term, which has the claim remain reciting *explicit interface*

member mechanism. See sec. 2.2.

As per claims 23-26, 28-37, 39-41: See related rationale addressed in claim 1-3, 6-15, 17-19.

As per claims 58-63, 66-75: See related rationale addressed in claim 1-3, 6-15, 17-19.

Conclusion

8.   Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Ted T. Vo whose telephone number is (571) 272-3706. The

examiner can normally be reached on 8:00AM to 4:30PM.

        If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, Wei Y. Zhen can be reached on (571) 272-3708.

The facsimile number for the organization where this application or proceeding is assigned is the

Central Facsimile number 571-273-8300.

        Any inquiry of a general nature or relating to the status of this application should be

directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of

an application may be obtained from the Patent Application Information Retrieval (PAIR)

system. Status information for published applications may be obtained from either Private PAIR

or Public PAIR. Status information for unpublished applications is available through Private

PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov.

Should you have questions on access to the Private PAIR system, contact the Electronic Business

Center (EBC) at 866-217-9197 (toll-free).

        TTV
        April 23, 2009

        /Ted T. Vo/
        Primary Examiner, Art Unit 2191